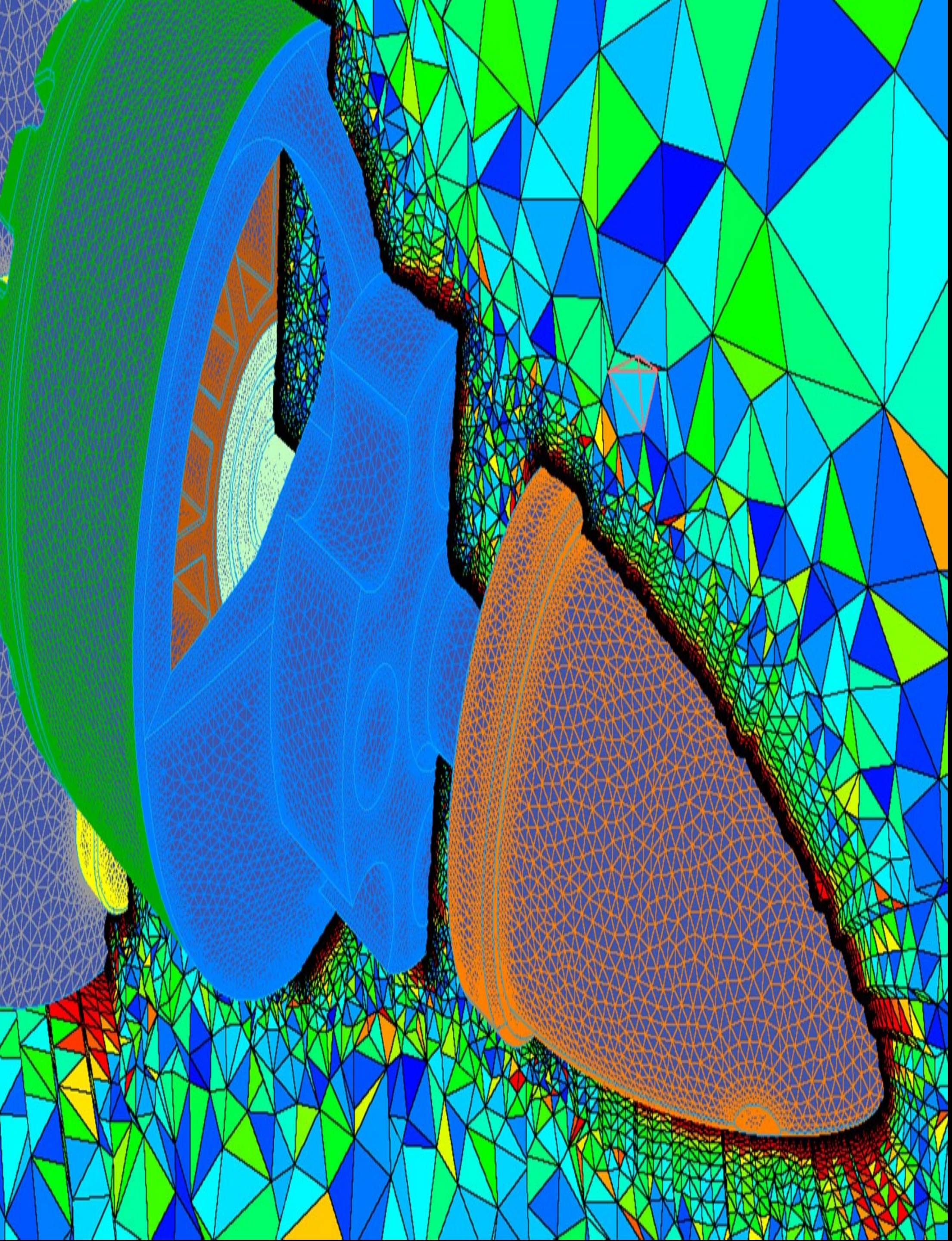


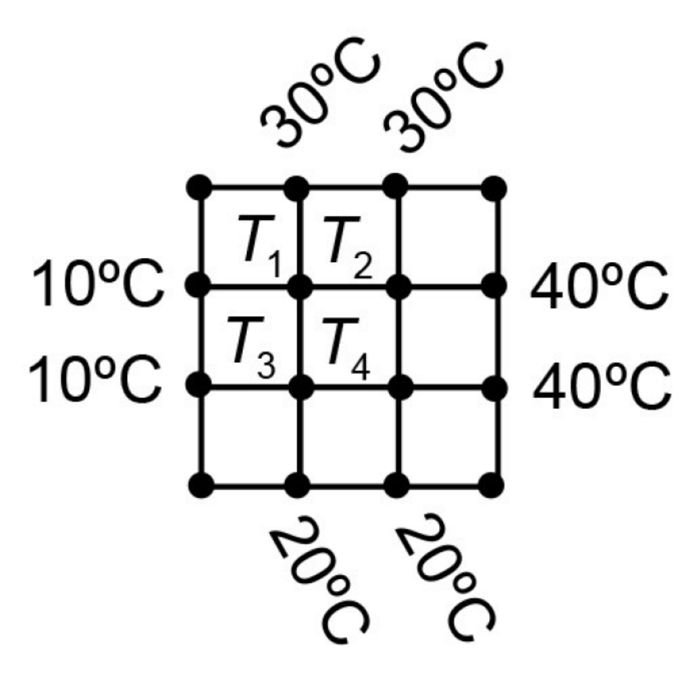
MÉTODO NUMÉRICOS

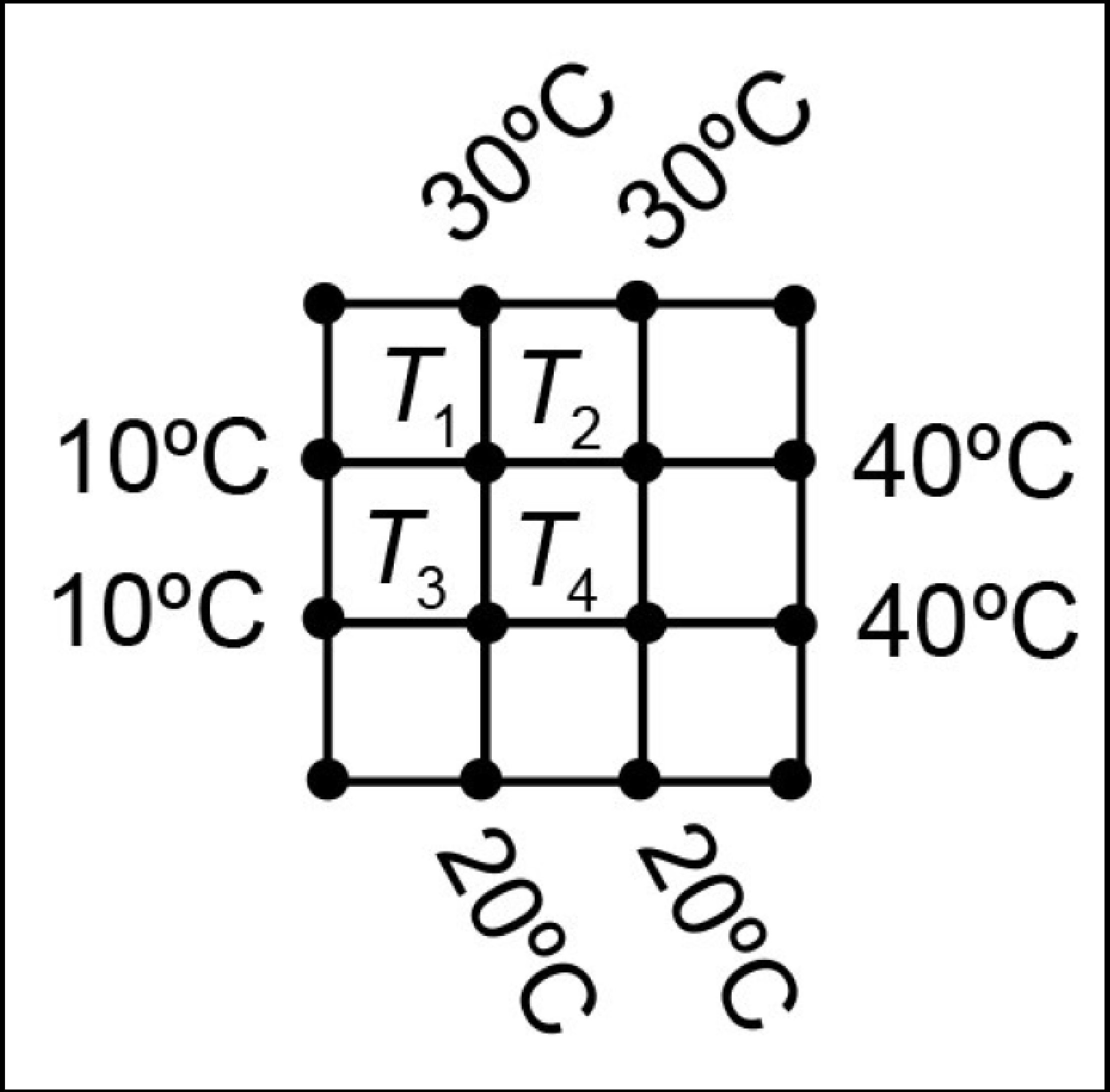
Aula 4 – Sistemas Lineares

Curso de Ciência da Computação
Dr. Rodrigo Xavier de Almeida Leão
Cientista de Dados

Os sistemas lineares possuem diversas aplicações em diferentes áreas do conhecimento, desde um simples problema de modelar um circuito elétrico até problemas complexos de computação científica, nas quais estão presentes sistemas lineares, cujas matrizes possuem milhares de linhas e colunas. O ponto de partida na resolução de um de um processador que opera em baixa temperatura. A Figura 2.1 ilustra a geometria de um processador e as respectivas temperaturas do seu contorno. Como pode ser observado, o bordo do processador possui diferentes faixas de temperatura, isso se deve ao fato de ele estar em contato com outros equipamentos; além disso, diferentes taxas de transferência de calor são obtidas pelo sistema de refrigeração para o processador.

um sistema linear, obtido a partir da seguinte lei: "a temperatura em determinado ponto é aproximadamente igual à média aritmética dos quatro pontos vizinhos a ele". A partir dessa lei, podemos obter as seguintes equações e, também, o sistema linear originado a partir delas:





Equação 1: $T_1 = \left(T_2 + 30 + 10 + T_3\right)/4$;

Equação 2: $T_2 = \left(40 + 30 + T_1 + T_4\right)/4$;

Equação 3: $T_3 = \left(T_4 + T_1 + 10 + 20\right)/4$;

Equação 4: $T_4 = \left(40 + T_2 + T_3 + 20\right)/4$;

$$\begin{pmatrix} 4 & -1 & -1 & 0 \\ -1 & 4 & 0 & -1 \\ -1 & 0 & 4 & -1 \\ 0 & -1 & -1 & 4 \end{pmatrix} \begin{pmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{pmatrix} = \begin{pmatrix} 40 \\ 70 \\ 30 \\ 60 \end{pmatrix}.$$

Sistemas lineares

A modelagem matemática de diversos fenômenos físicos, biológicos ou químicos resulta no problema de resolver simultaneamente diversas equações lineares. Em linguagem matemática, esses fenômenos são descritos por um conjunto de m equações, no qual se deseja determinar a solução de n variáveis denominadas incógnitas. A esse conjunto de equações damos o nome de sistema linear. Cada uma das equações que compõem esse sistema é definida por:

Exemplificando

Considere a equação linear $6x_1 - 2x_2 + x_3 = 3$. O vetor $\overline{x} = (1,1,-1)$ é a solução dessa equação, uma vez que a substituição deste na equação linear torna a igualdade consistente: $6 \cdot 1 - 2 \cdot 1 + 1 \cdot (-1) = 3$.

Definição 2: um sistema de m equações lineares com n incógnitas $(m,n \geq 1)$ é um conjunto de m equações lineares, cada uma delas com n incógnitas avaliadas simultaneamente. Matematicamente, é escrito como:

$$S:\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \qquad\qquad\vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = b_m \end{cases}$$

Em S os coeficientes são reais, $a_{i,j}, b_i \in \mathbb{R}$ para todo $i = 1,\ldots,m$ e $j = 1,\ldots,n$, e resolver esse sistema linear equivale a encontrar $\overline{x} = \left(\overline{x}_1, \overline{x}_2, \ldots, \overline{x}_n\right)$ tal que todas as equações do sistema sejam satisfeitas simultaneamente.

Uma representação bastante útil de S é sua forma matricial dada por $A_{m\times n}x_n = b_m$, onde $A_{m\times n}$ é a matriz dos coeficientes, x_n o vetor de incógnitas e x_n o vetor dos termos independentes, respectivamente denotados por:

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix}, \qquad x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}, \qquad b = \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix}.$$

Considere o sistema linear $\begin{cases} 7x_1 - 7x_2 + 7x_3 = 7 \\ 4x_1 + 2x_2 + 4x_3 = 0 \\ 3x_1 - 1x_2 + 1x_3 = 1 \end{cases}$. A forma matricial associada a ele é dada por:

$$Ax = b \Rightarrow \begin{pmatrix} 7 & -7 & 7 \\ 4 & 2 & 4 \\ 3 & -1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 7 \\ 0 \\ 1 \end{pmatrix} \Rightarrow A = \begin{pmatrix} 7 & -7 & 7 \\ 4 & 2 & 4 \\ 3 & -1 & 1 \end{pmatrix}, \ x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}, \ b = \begin{pmatrix} 7 \\ 0 \\ 1 \end{pmatrix}$$

Outra representação matricial importante dos sistemas lineares é conhecida como matriz aumentada. Nesta representação, os elementos $a_{i,j}$ e b_j estão reunidos em uma única matriz de dimensão $m \times (n+1)$, conforme segue:

$$\left(\begin{array}{ccc|c} a_{11} & \cdots & a_{1n} & b_1 \\ \vdots & \ddots & \vdots & \vdots \\ a_{m1} & \cdots & a_{mn} & b_m \end{array}\right).$$

Essa forma de representar o sistema linear é bastante útil nas iterações de diversos métodos numéricos. Por fim, é importante observar que sempre que nos referirmos, nesta Unidade 2, a um sistema linear na forma $A_{m \times n} x = b_m$, estaremos considerando um sistema linear quadrado, ou seja, aquele em que o número de equações é igual ao número de incógnitas, portanto, $m = n$.

Definição: sejam A e B matrizes de dimensão $m \times n$, então a soma dessas duas matrizes é uma matriz C de dimensão $m \times n$, cujas entradas são $c_{i,j} = a_{i,j} + b_{i,j}$ para cada $i = 1, \cdots, m$ e $j = 1, \ldots, n$.

Exemplificando

Considere as seguintes matrizes $A = \begin{bmatrix} 1 & 5 & 9 \\ 3 & 2 & 1 \\ 7 & 4 & 3 \end{bmatrix}$ e $B = \begin{bmatrix} 0 & 4 & 1 \\ 3 & 5 & 2 \\ 1 & 6 & 2 \end{bmatrix}$ e determine $C = A + B$. Uma vez que as entradas da matriz C são dadas por $c_{i,j} = a_{i,j} + b_{i,j}$, então tem-se:

$$C = \begin{bmatrix} a_{11} + b_{11} & a_{12} + b_{12} & a_{13} + b_{13} \\ a_{21} + b_{21} & a_{22} + b_{22} & a_{23} + b_{23} \\ a_{31} + b_{31} & a_{32} + b_{32} & a_{33} + b_{33} \end{bmatrix} = \begin{bmatrix} 1 & 9 & 10 \\ 6 & 7 & 3 \\ 8 & 10 & 5 \end{bmatrix}.$$

Calculando o produto $\lambda \cdot A$ sendo $A = \begin{bmatrix} 3 & 1 & 5 \\ 2 & 7 & 1 \\ 1 & 3 & 4 \end{bmatrix}$ e $\lambda = 2$, temos:

$$\lambda \cdot A = 2 \cdot \begin{bmatrix} 3 & 1 & 5 \\ 2 & 7 & 1 \\ 1 & 3 & 4 \end{bmatrix} = \begin{bmatrix} 6 & 2 & 10 \\ 4 & 14 & 2 \\ 2 & 6 & 8 \end{bmatrix}.$$

Definição (produto entre matriz e vetor): seja A uma matriz de dimensão $m \times n$ e x um vetor de dimensão n. O produto entre a matriz A e o vetor x é denotado por $A \cdot x$ e matematicamente calculado como a seguir:

$$A \cdot x = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} \sum_{j=1}^{n} a_{1,j} \cdot x_j \\ \sum_{j=1}^{n} a_{2,j} \cdot x_j \\ \vdots \\ \sum_{j=1}^{n} a_{m,j} \cdot x_j \end{bmatrix}.$$

O produto entre a matriz $A = \begin{bmatrix} 8 & 3 & 5 \\ 2 & 4 & 1 \\ 1 & 3 & 2 \end{bmatrix}$ e o vetor $\boldsymbol{x} = \begin{bmatrix} 2 \\ 3 \\ 1 \end{bmatrix}$ tem como resultado a matriz apresentada a seguir:

$$A \cdot b = \begin{bmatrix} 8 & 3 & 5 \\ 2 & 4 & 1 \\ 1 & 3 & 2 \end{bmatrix} \cdot \begin{bmatrix} 2 \\ 3 \\ 1 \end{bmatrix} = \begin{bmatrix} 8 \cdot 2 + 3 \cdot 3 + 5 \cdot 1 \\ 2 \cdot 2 + 4 \cdot 3 + 1 \cdot 1 \\ 1 \cdot 2 + 3 \cdot 3 + 2 \cdot 1 \end{bmatrix} = \begin{bmatrix} 30 \\ 17 \\ 13 \end{bmatrix}.$$

Definição (matriz identidade): uma matriz identidade de ordem n, denotada por I_n, é uma matriz cuja diagonal principal, ou seja, os elementos $a_{i,i} = 1$ para $i = 1, \ldots, n$ e todos os demais são zeros.

Definição (matriz transposta): a transposta de uma matriz $A = \left[a_{i,j}\right]$ de dimensão $m \times n$ é a matriz $A^T = \left[a_{j,i}\right]$ de dimensão $n \times m$, em que os elementos de cada linha i de A^T são ordenadamente iguais aos elementos da coluna i de A.

Matriz identidade

$$I_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A = \begin{bmatrix} 1 & 5 \\ 3 & 2 \\ 4 & 12 \\ 7 & 9 \end{bmatrix} \qquad \boxed{A^T} = \begin{bmatrix} 1 & 3 & 4 & 7 \\ 5 & 2 & 12 & 9 \end{bmatrix}$$

```python
import numpy as np

# Criando duas matrizes
A = np.array([[1, 2], [3, 4]])
B = np.array([[5, 6], [7, 8]])

# Adição de matrizes
C = A + B
print("Adição de Matrizes:\n", C)

# Subtração de matrizes
D = A - B
print("\nSubtração de Matrizes:\n", D)

# Multiplicação por um escalar
E = 2 * A
print("\nMultiplicação por Escalar:\n", E)

# Multiplicação de matrizes
F = np.dot(A, B)  # ou A @ B
print("\nMultiplicação de Matrizes:\n", F)

# Transposta de uma matriz
G = A.T
print("\nTransposta de Matriz:\n", G)

```

```python
# Inversa de uma matriz
try:
  H = np.linalg.inv(A)
  print("\nInversa de Matriz:\n", H)
except np.linalg.LinAlgError:
  print("\nMatriz não possui inversa.")

# Determinante de uma matriz
I = np.linalg.det(A)
print("\nDeterminante de Matriz:\n", I)
```

APLICAR OS OPERADORES MATRICIAIS NAS MATRIZES ABAIXO

```
[[  6   6  -1   4   6]
 [  9  -1   0   0   4]
 [  5 -10  -6   4  -5]
 [ -1  -5  -7   6   3]
 [  5   7   3   3   9]]
```

```
[[ -4   9   6   9   7]
 [ -6  -9  -1   7   4]
 [ -2  -3  10  -2 -10]
 [  4  -4  -3  -8  -8]
 [ -6   9  -3  -1   2]]
```

```python
import numpy as np

def solve_linear_system(A, b):

  try:
    x = np.linalg.solve(A, b)
    return x
  except np.linalg.LinAlgError:
    print("O sistema linear não tem solução única.")
    return None

# Exemplo de uso:
A = np.array([[ 1, -2 ,-7],
              [ 8 ,-1 ,-7],
              [-1  ,6  ,8]])

b = np.array([1, 0, 2])

x = solve_linear_system(A, b)

if x is not None:
  print("Solução do sistema:")
  print(x)
```

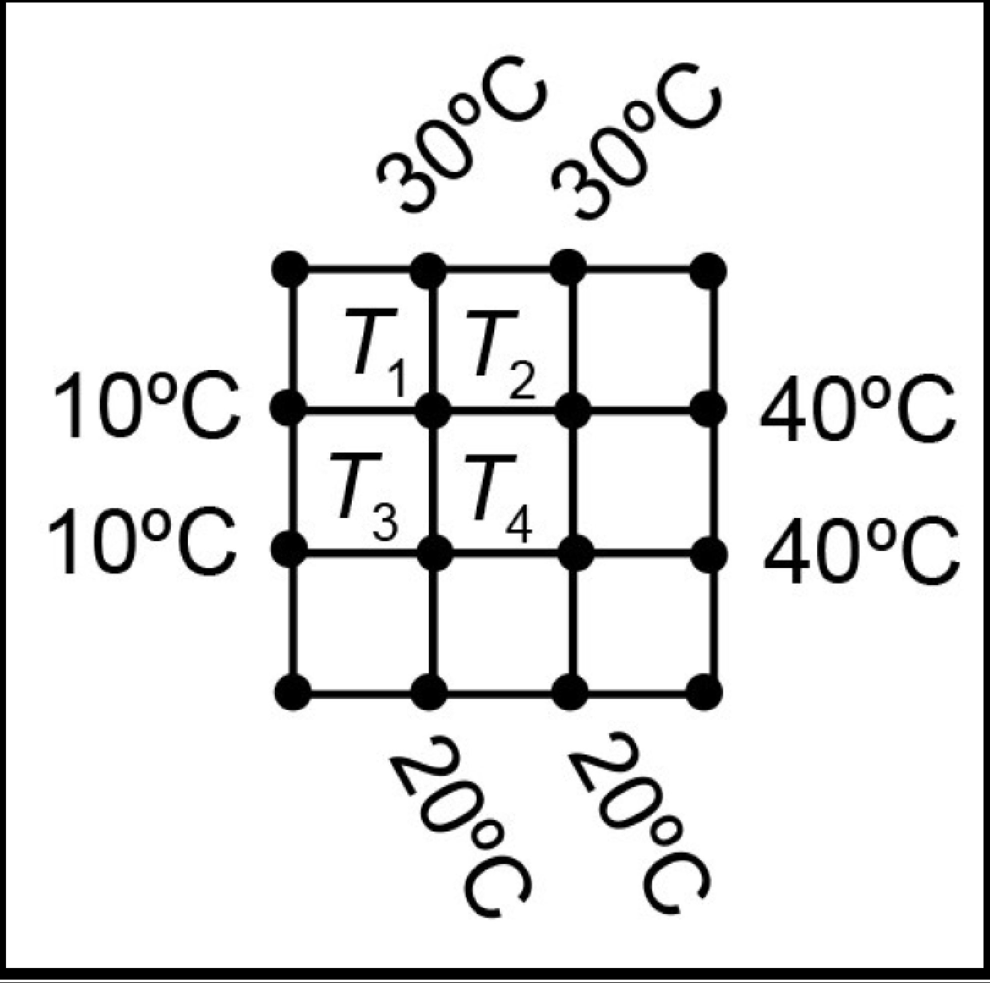
SOLUCIONAR OS SISTEMAS

$$\begin{cases} 7x_1 - 7x_2 + 7x_3 = 7 \\ 4x_1 + 2x_2 + 4x_3 = 0 \\ 3x_1 - 1x_2 + 1x_3 = 1 \end{cases}$$

$$\begin{pmatrix} 4 & -1 & -1 & 0 \\ -1 & 4 & 0 & -1 \\ -1 & 0 & 4 & -1 \\ 0 & -1 & -1 & 4 \end{pmatrix} \begin{pmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{pmatrix} = \begin{pmatrix} 40 \\ 70 \\ 30 \\ 60 \end{pmatrix}$$

```python
import matplotlib.pyplot as plt
import numpy as np

def gerar_malha_de_calor(matriz):
  """
  Gera uma malha de calor de uma matriz do vermelho para o azul.

  Args:
    matriz: A matriz de valores para gerar a malha de calor.
  """

  plt.imshow(matriz, cmap='RdBu')
  plt.colorbar()
  plt.show()

# Exemplo de uso:
matriz =  np.array([[ 1, -2 ,-7],
              [ 8 ,-1 ,-7],
              [-1  ,6  ,8]]) # Matriz de exemplo
gerar_malha_de_calor(matriz)
```

GERAR O MAPA DE CALOR



Método de Gauss

Neste ponto, temos todo o ferramental teórico matemático necessário para a implementação do método de eliminação de Gauss. Como apresenta Ruggiero (2005), o método de Gauss consiste em transformar o sistema linear original $Ax = b$ em um sistema linear equivalente $Ax = b$, denominado triangular superior.

Assimile

Denomina-se sistema triangular superior todo sistema $Ax = b$ em que $a_{i,j} = 0$ para qualquer $i > j$, ou seja:

$$\begin{cases} a_{11}x_1 & + & a_{12}x_2 & + & \cdots & + & a_{1n}x_n & = & b_1 \\ 0x_1 & + & a_{22}x_2 & + & \cdots & + & a_{2n}x_n & = & b_2 \\ \vdots & & \vdots & & & & \vdots & & \vdots \\ 0x_1 & + & 0x_2 & + & \cdots & + & a_{n,n}x_n & = & b_n \end{cases} \Rightarrow \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ 0 & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{n,n} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$

A solução de um sistema triangular superior é facilmente obtida por um processo de retrossubstituição, que algoritmicamente é descrito por:

$$x_n = \frac{b_n}{a_{n,n}}$$

para $i = (n-1), \ldots, 1$ faça

$soma = 0.0$;

para $j = (i+1), \ldots, n$ faça

$soma = soma + a_{i,j} x_j$;

fim_para

$x_i = (b_i - soma)/a_{ii}$;

fim_para

Utilize o método de Gauss para a resolução do sistema linear a seguir.

$$\begin{pmatrix} 5 & 3 & 4 \\ 4 & 2 & 3 \\ 2 & 1 & 2 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 4 \\ 2 \\ 1 \end{pmatrix} \Rightarrow A^{(0)} \mid b^{(0)} = \left(\begin{array}{ccc|c} a_{1,1}^{(0)} & a_{1,2}^{(0)} & a_{1,3}^{(0)} & b_1^{(0)} \\ a_{2,1}^{(0)} & a_{2,2}^{(0)} & a_{2,3}^{(0)} & b_2^{(0)} \\ a_{3,1}^{(0)} & a_{3,2}^{(0)} & a_{3,3}^{(0)} & b_3^{(0)} \end{array}\right) = \left(\begin{array}{ccc|c} 5 & 3 & 4 & 4 \\ 4 & 2 & 3 & 2 \\ 2 & 1 & 2 & 1 \end{array}\right)$$

Resolução:

Etapa 1) Nesta etapa, o pivô é $a_{1,1}^{(0)} = 5$ e os multiplicadores são $m_{2,1} = \dfrac{a_{2,1}^{(0)}}{a_{1,1}^{(0)}} = \dfrac{4}{5} = 0{,}8$ e $m_{3,1} = \dfrac{a_{3,1}^{(0)}}{a_{1,1}^{(0)}} = \dfrac{2}{5} = 0{,}4$. Considerando as operações elementares, $a_{2,1}^{(0)}$ é eliminado por $L_2 = L_2 - m_{2,1}L_1$, e $a_{3,1}^{(0)}$ por $L_3 = L_3 - m_{3,1}L_1$; assim, tem-se:

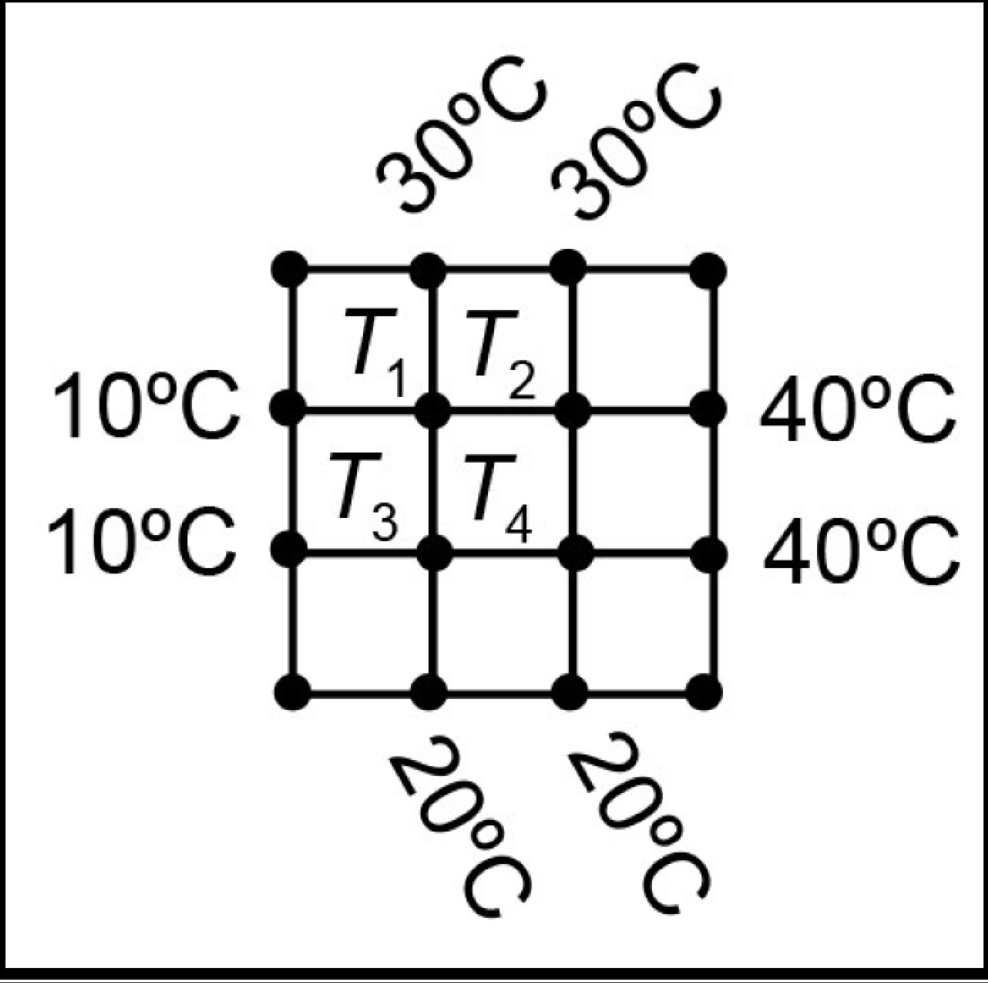
$$A^{(1)} \mid b^{(1)} = \left(\begin{array}{ccc|c} a_{1,1}^{(1)} & a_{1,2}^{(1)} & a_{1,3}^{(0)} & b_1^{(1)} \\ 0 & a_{2,2}^{(1)} & a_{2,3}^{(0)} & b_2^{(1)} \\ 0 & a_{3,2}^{(1)} & a_{3,3}^{(0)} & b_3^{(1)} \end{array}\right) = \left(\begin{array}{ccc|c} 5 & 3 & 4 & 4 \\ 0 & -0{,}4 & -0{,}2 & -1{,}2 \\ 0 & -0{,}2 & 0{,}4 & -0{,}6 \end{array}\right)$$

```python
import numpy as np

def gauss_elimination(A, b):
  """
  Resolve um sistema linear usando el
  Args:
    A: A matriz de coeficientes do si
    b: O vetor de termos independente
    """
  n = len(b)

  # Criar uma cópia das matrizes A e
  A_copy = np.copy(A).astype(float) #
  b_copy = np.copy(b).astype(float) #
```

```python
# Eliminação de Gauss para transformar a matriz em triangular s
for i in range(n - 1):
    for j in range(i + 1, n):
        if A_copy[i, i] == 0:
            # Pivotamento: troca linhas se o elemento diagonal
            for k in range(i + 1, n):
                if A_copy[k, i] != 0:
                    A_copy[[i, k]] = A_copy[[k, i]]
                    b_copy[[i, k]] = b_copy[[k, i]]
                    break
            else:
                # If no suitable pivot is found, the matrix is
                raise np.linalg.LinAlgError("Singular matrix")

        factor = A_copy[j, i] / A_copy[i, i]
        A_copy[j, i:] = A_copy[j, i:] - factor * A_copy[i, i:]
        b_copy[j] = b_copy[j] - factor * b_copy[i]

# Resolução por substituição retroativa
x = np.zeros(n)
x[n - 1] = b_copy[n - 1] / A_copy[n - 1, n - 1]
for i in range(n - 2, -1, -1):
    sum_term = np.dot(A_copy[i, i+1:], x[i+1:])
    x[i] = (b_copy[i] - sum_term) / A_copy[i, i]

return A_copy, b_copy, x
```

```python
  # Exemplo de uso:
A = np.array([[5, 3, 4],
              [4, 2, 3],
              [2, 1, 2]])

b = np.array([4, 2, 1])

print("Sistema original:")
print("A:\n", A)
print("b:\n", b)

A_triangular, b_modified, solution = gauss_elimination(A, b)

print("\nMatriz triangular superior:")
print(A_triangular)
print("\nVetor de termos independentes:")
print(b_modified)
print("\nSolução do sistema:")
print(solution)
```

SOLUCIONE O SISTEMA UTILIZANDO O MÉTODO DE GAUSS E COMPARE COM O MÉTODO ANTERIOR



```
Sistema original:
A:
 [[5 3 4]
 [4 2 3]
 [2 1 2]]
b:
 [4 2 1]

Matriz triangular superior:
[[ 5.   3.   4. ]
 [ 0.  -0.4 -0.2]
 [ 0.   0.   0.5]]

Vetor de termos independentes:
[ 4.  -1.2  0. ]

Solução do sistema:
[-1.  3.  0.]
```